A method of searching LDAP directories using XQuery

Ted Hesselroth

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510-5011

E-mail: tdh@fnal.gov

Abstract. A method by which an LDAP directory can be searched using XQuery is described. The strategy behind the tool consists of four steps. First the XQuery script is examined and relevant XPath expressions are extracted, determined to be sufficient to define all information needed to perform the query. Then the XPath expressions are converted into their equivalent LDAP search filters by use of the published LDAP schema of the service, and search requests are made to the LDAP host. The search results are then merged and converted to an XML document that conforms to the hierarchy of the LDAP schema. Finally, the XQuery script is executed on the working XML document by conventional means. Examples are given of application of the tool in the Open Science Grid, which for discovery purposes operates an LDAP server that contains Glue schema-based information on site configuration and authorization policies. The XQuery scripts compactly replace hundreds of lines of custom python code that relied on the unix ldapsearch utility. Installation of the tool is available through the Virtual Data Toolkit.

1. Information Services and the Grid

Grid computing is the practice of utilizing computing resources that are deployed across administrative domains[1]. In simpler terms, it is performing computations at multiple sites which are operated by different institutions. One premise of grid computing is that sites are configured at the discretion of their operators; therefore, sufficient information to use the resource is not available a priori. Even the presence of resources may change over time.

Grid computation requires interaction within this dynamic and heterogeneous environment. Having accurate and timely information on the deployed infrastructure is an essential component of effective grid utilization, not only for the discovery of services but for uncovering their configuration, without which it may be impossible to actually use them.

Due to the dynamic nature of independently-operated sites, a centralized information service was chosen by the Worldwide LHC Computing Grid (WLCG) [2] to support grid computing for the Large Hadron Collider. The service data store is populated by WLCG sites, which transmit information on installed services and their configuration to the central server. The information thus becomes available to clients which query the server. The use of information services for discovery has precedents in previous computing models, such as DNS service discovery, UDDI, Bluetooth, WS-Discovery, Web Proxy autodiscovery, and DHCP. One of the prominent ideas involving discovery is that of the "Semantic Web", put forward by Tim Berners-Lee[3]. Such a model is based on the concept of machine-readable metadata. For example, the Resource Description Framework and the Web Ontology Framework have been used to facilitate the exchange of information between services by describing the information content at the metadata level.

1.1. The Lightweight Directory Access Protocol

Of course, there are a variety of ways of serving information on the internet, besides the ubiquitous HTML format delivered using the http protocol. The method chosen by WLCG is to use LDAP, the Lightweight Directory Access Protocol[4]. LDAP offers various advantages, not the least of which is an available implementation from the University of Michigan. Features such as secure socket communication, authorization and access control lists, and the use of URLS to specify servers are available.

The origin of LDAP is in X.500 databases, early forms of directories. X.500 databases were first discussed in 1984 and standardized in 1990. Most uses of X.500 databases are for convenient lookup of simple information such as email directories, network IP address books, public key repositories, and printer names. LDAP, which was standardized in 1993, originally used X.500 as its backing database. As of version 3, besides X.500 in the form of the "slapd" service, Novell Directory Service or Microsoft Active Directory may alternatively be used.

Besides the actual protocol for exchanging directory information, LDAP provides the ability to define an information model, or schema, customized to the information being presented. This allows the delivery of information about anything for which a schema can be devised. Another part of LDAP is LDIF, the LDAP Interchange Format. The format specifies a consistent way of presenting of LDAP entries and their attributes as text headers and name/value pairs, allowing for serialization and the exchange of information between LDAP clients and servers.

The actual software used as the information server is the Berkeley Database Information Index[5]. BDII is an LDAP server with the additional capability of aggregating information from multiple sources. In practice, sites deliver information to the BDII server in the form of LDIFs, which BDII then merges and uses to update the database. The LDIF information from the sites is gathered by Generic Information Providers[6], which either read information from manually-maintained files or obtain it from resource-specific probes installed alongside their respective service. The LDIF-encoded information is delivered to BDII via the CEMon[7] utility. BDII servers are run by both the WLCG in Europe and by the Open Science Grid[8] in the United States. The WLCG BDII service is periodically updated by all of the information in its OSG counterpart.

1.2. The Glue Schema

The schema used for the WLCG is called the Glue Schema[9], named for the Grid Laboratory Uniform Environment. Work on the glue schema had begun in 2002, in a collaboration between the EU-DataTAG and US-iVDGL projects. The version supported by the software described in this paper is 1.3. The Glue Schema serves as an abstract model of a site environment. It is informed by the features of software commonly found on grid sites, with sufficient coverage of services and their parameters such that the resources that a site chooses to make available may be advertised.

The glue schema is organized in a manner similar to that of a site's system architecture. For example, there is a high-level class for a Computing Element. There is also a high-level class for a Storage Element, which has a subclass Access Protocol that describes the means by which data may be moved to and from the Storage Element. Examples of the latter are gridftp and xrootd. The class Access Protocol, in turn, has an attribute named Endpoint, that is, the URL by which to contact the service. Another subclass of Storage Element is Storage Area, corresponding to the partitioning a Storage Element into parts of different sizes. A Storage Area may have its own subclass called VO Info, describing which Virtual Organizations are allowed to access it, and by what paths. These are just examples of the many types and layers of information found in the Glue Schema.

1.3. XQuery vs LDAP filters

One drawback of the use of LDAP in the WLCG is the very complexity of the Glue Schema. While in principle the LDAP service itself can support quite complex schemas, in practice the schemas have been relatively simple. Consequently, clients have not needed the capacity to perform complex searches. LDAP is searched using "filters", defined in RFC2254[10], which do provide for limited selection of the search object and attribute names along with basic comparison and Boolean operators. However, no exploitation of any structured data model represented in the schema is supported by LDAP filters. Furthermore, in common search tools such as ldapsearch, results are returned as LDIF, requiring custom parsing tools for extraction of the values of interest. In practice, complex LDAP searches must be composed of multiple simple filter searches, with intermediate parsing of results and reformulation of filter parameters. With reasonable expertise, this can be achieved with a scripting language such as shell script or Python, but results in search tools that are quite specialized in their purpose and would require learning of a particular data handling motif to modify or extend.

Fortunately, the Glue Schema, though expressible as LDAP schema, is not dependent on LDAP. Since it is a hierarchical arrangement of elements and attributes, the Glue Schema may also be expressed as an XML schema. Any informational item that exists in the LDAP directory thus has a corresponding XPath expression associated with it. This suggests the possibility of performing searches based not on LDAP filters, but on XPath[11] expressions, and even further, composing more complex searches using XQuery[12]. One advantage of such an approach would be that it would not be necessary to learn the particulars of some scripting framework based on LDAP search filters (if one existed) in order to write complex searches. It would still be necessary to know about XPath and XQuery, but that would represent the acquisition of knowledge of rather general use, which would likely benefit the practitioner. Furthermore, these languages are designed precisely for the extraction and manipulation of hierarchical data, and therefore the expressions in case of deep schemas such as the Glue Schema, are more natural to work with.

Another advantage of using XQuery as the search motif is the widespread use of XML in other information services. Since interoperability is a design principle of grid computing, interoperability within the Information Services domain itself is a valuable feature. Today, XML is the de facto standard for informational interoperability. Basing the search technology on XQuery anticipates possible future integration or replacement of LDAP with other information services based on XML.

1.4. XML and Directory Services

XML, the Extensible Markup Language[13], is based on Standard Generalized Markup Language[14], that is, it is a specific profile of SGML (just as HTML is). SGML is a specification for defining markup languages that was adopted as an ISO standard in 1986. Work on the XML profile was begun in 1996 and the language became a W3C standard in 1998. XML is a based on documents: specific collections of information constructed according to the rules of the language. The main feature of XML is the ability to define a schema, which describes the informational encoding within a document. Markup within the document itself follows the definition of the schema and allows the extraction of specific information. XML provides further features, such as the specification of character encoding and references to other documents.

Various derivative technologies have been developed based on the XML standard. XPath[11] is a W3C specification from 1999 and 2007 for describing navigation within an XML document, canonically by the construction of the path to an item of information within the document tree, made by the composition of the names of its ancestor elements. XPath also provides for more abstract expressions that extract information from the document based on comparison, Boolean, or arithmetic operations on elements and attributes. Thus, it is common to say that and XPath expression is "evaluated" on a document.

Other technologies that are, in turn, based on XPath are XSLT and XQuery. XSLT, or Extensible Stylesheet Language Transformations[15], is a language for transforming XML documents. The transformations are themselves written as XML documents according to the schema as part of its W3C 1999 standard. XQuery, the Extensible Query Language[13], which became a W3C standard in 2007, is similar to XSLT in capability, but with a style more similar to procedural programming. As its name suggests, its constructs are suitable for the application of performing queries on a document. Both languages rely on XPath, but the XQuery language more explicitly so.

By defining a schema in XML, a markup language for a specific purpose may be created. Many of these have been specified since the inception of XML, such as XHTML for expressing web pages, MathML for mathematical objects, OpenDocument for office documents, SOAP for web service messages, SAML and XACML for authorization software, etc. A markup language which enables the use of XQuery to search LDAP directories is the Directory Services Markup Language (DSML).

The Directory Services Markup Language[16] is an XML schema that allows LDAP data to be expressed in XML format. This provides an alternative to the LDIF representation described above. DSML became an OASIS standard in 2001. The standard also includes schema for performing other operations on LDAP servers, such as updates and deletes. DSML Tools[17]provided, and continues to maintain Java-based tools for performing LDAP searches and outputting the results as XML.

2. An XQuery Strategy for searching LDAP

Based on the ability of expressing Glue Schema as XML schema, and the availability of a tool to acquire LDAP search results as XML, the following strategy for searching an LDAP directory using XQuery presents itself:

1. Write the XQuery script, based on the Glue Schema.
2. Examine the XQuery script and extract the XPaths from it.
3. Convert the XPath expressions to LDAP filters.
4. Perform searches on the ldap host using the ldap filters, and merge results as XML file.
5. Run the XQuery on the resulting XML file.

The following diagram illustrates the step of the process, and the associated software components.

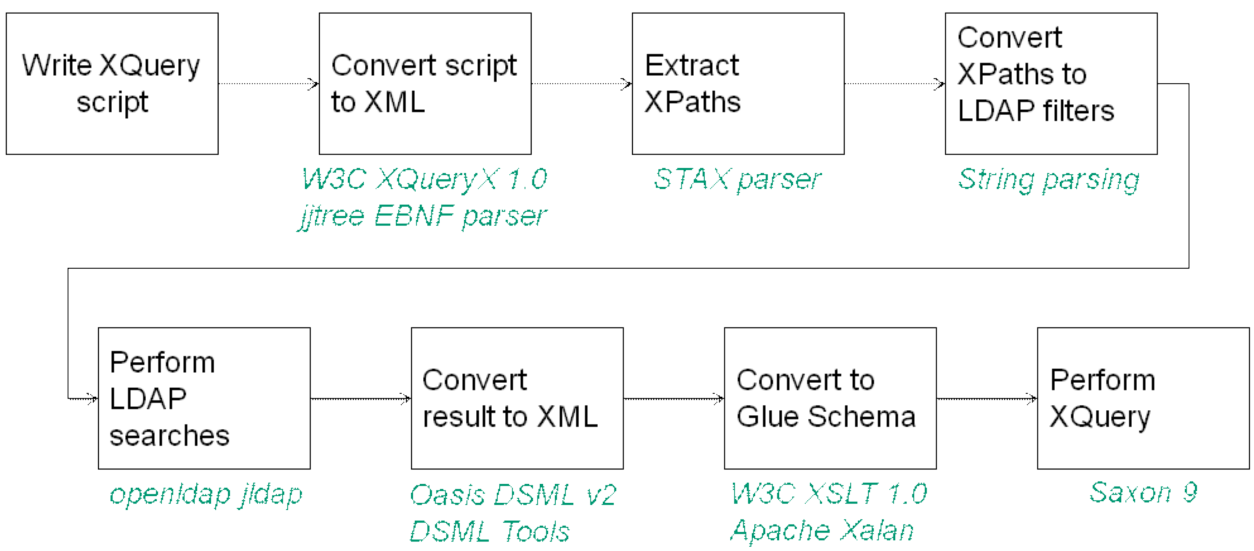

Figure 1. Diagram of the processing steps.

Details related to these steps are as follows.

2.1. The XQuery scripts.

While one of the goals of the software was to make it easier to write queries for the directory service, several XQuery scripts thought to be of particular value to grid users were written and bundled together with the software to provide a tool. The so-named OSG Discovery Tool[18] provides search capabilities runnable through wrapper scripts by which the user can provide search parameters so as to obtain customized results without the need for any programming. The tool is an Open Science Grid[8]software product and is provided through the Virtual Data Toolkit[19] package.

2.2. Parsing the XQuery.

Since the XQuery is itself an XML document, XPaths are extracted from it by means of a parsing operation. The XQuery is preprocessed into XQueryX[20] form (W3C standard) in order to facilitate recognition of the location axes and step expressions. Additional elements such as comparison and Boolean operators could also be detected. A STAX[21] processor was used to create lists of such elements, which were then used to construct the XPath expressions.

2.3. The LDAP filters.

LDAP filters were constructed from the XPath expressions by parsing them as strings. Since both XPath expressions and LDAP filters can include comparison and Boolean operations, the XPath expressions were constructed so as to include them if present, and the operations carried forward to the LDAP filters. This allows a more precise search on the LDAP directory, reducing the volume of XML data to be later queried.

2.4. Performing the LDAP search.

The openldap[22] jldap software is used to perform the search, and the result transformed to XML using DSML Tools. The result, while in XML, is expressed in generic DSML schema rather than Glue Schema. Therefore, this step includes an additional process of converting the result to Glue Schema by means of an XSL transform.

2.5. Running the XQuery.

The Saxon[23]software is used to perform the XQuery on the resulting XML. In the Discovery Tools software package, in order to avoid the invocation of steps 1-4 every time a command is run, one XML file that is sufficient for all provided XQueries is used. The file is kept current; when a user performs a search, if the file has expired it is refreshed. This is done by extracting the XPaths from all XQueries provided with the tool and performing steps 3 and 4 on that list.

3. The OSG Discovery Tool

In the OSG Discovery Tool package, the processing steps are executed by shell scripts, started off by a wrapper shell script named for the particular XQuery to be executed. For example an XQuery, which finds the Storage Elements supporting a particular Virtual Organization and constructs a site URL for each one, has a wrapper named get_surl. An example invocation and output are shown below.

```
[discovery-1.0.6]$ get_surl --vo jdem

srm://bsrm-1.t2.ucsd.edu:8443/srm/v2/server?SFN=/hadoop/jdem/TESTFILE
srm://fndca1.fnal.gov:8443/srm/managerv2?SFN=/pnfs/fnal.gov/usr/fermigrid/jdem/TESTFILE
srm://osg.crc.nd.edu:8443/srm/v2/server?SFN=/dscratch/osg/bestman/TESTFILE
srm://osg-se.sprace.org.br:8443/srm/managerv2?SFN=/pnfs/sprace.org.br/data/TESTFILE
```

Figure 2. A get_surl example.

Formatting is also supported. In the get_storage_area command, an XQuery is called which returns several pieces of information on advertised Storage Areas. It is convenient to display this output in columns, as shown in the following example. An HTML table format is also available.

```
[discovery-1.0.6]$ get_storage_area

STORAGE ELEMENT ID        STORAGE AREA NAME              TOTAL     FREE      VOs
atlas.bu.edu              Default Storage Area           122984    26264     atlas,cdf,mis,ops,osg
atlas07.cs.wisc.edu       Default Storage Area           0         0         atlas
bsrm-1.t2.ucsd.edu        Default Storage Area           1010185   100463    accelerator,astro,atlas,…
charm.ucr.edu             Default Storage Area           6730      1388      atlas,cdf,cms,engage,…
cit-se.ultralight.org     Default Storage Area           526869    109617    DOSAR,LIGO,alice,atlas,…
cmssrm.fnal.gov           flushPools:replica:online      6984011   1870180   cms,mis
cmssrm.fnal.gov           LFSOnlyPools:replica:online    529440    377902    cms,mis
cmssrm.fnal.gov           ResilientPools:replica:online  428079    195002    cms,mis
cmssrm.fnal.gov           stagePools:custodial:nearline  161695    37560     cms,mis
cmssrm.hep.wisc.edu       black-pools:replica:online     20186     11618     LIGO,atlas,cdf,cms,…
…
```

Figure 3. A get_storage_area example, showing columnar formatting.

4. Acknowledgments

This work was supported by the Office of Science, U.S. Department of Energy, SciDAC program under Contract DE-FC02-06ER41436 and the National Science Foundation Cooperative Agreement, PHY-0621704. Fermilab is operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy. Fermilab is partially funded by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Dept. of Energy, under Contract DE-AC02-06CH11357.

References

[1] Foster, Ian; Carl Kesselman 1999 The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers. ISBN 1-55860-475-8.
[2] The Worldwide LHC Computing Grid (WLCG), http://lcg.web.cern.ch/LCG.
[3] T. Berners-Lee, T, J. Hendler O. Lassila 2001 The Semantic Web. Sci. American Magazine.
[4] Howes, Timothy A. 1997 The Lightweight Directory Access Protocol : X.500 Lite. CITI Technical Report 95–8. http://www.openldap.org/pub/umich/ldap.pdf.
[5] BDII web site. https://twiki.cern.ch/twiki/bin/view/EGEE/BDII.
[6] Czajkowski, K., Fitzgerald, S., Foster, I. And Kesselman, C. 2001 "Grid information services for distributed resource sharing," 10th IEEE International Symposium on High Performance Distributed Computing. Proceedings. pp.181-194.
[7] CEMON home page. http://grid.pd.infn.it/cemon/field.php.
[8] Open Science Grid web site. http://www.opensciencegrid.org.
[9] Glue Schema web site. http://glueschema.forge.cnaf.infn.it.
[10] Howes, T. 1997 The String Representation of LDAP Search Filters. RFC 2254.
[11] W3C XPath web site. http://www.w3.org/TR/xpath.
[12] W3C XQuery web site. http://www.w3.org/XML/Query.
[13] W3C XML web site. http://www.w3.org/XML.
[14] P. Kilpel and D. Wood 2001 SGML and XML document grammars and exceptions. Inf. Comput. 169, 2.
[15] W3C XSLT web site. http://www.w3.org/Style/XSL.
[16] OASIS DSMl web site. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=dsml.
[17] DSML Tools web site. http://www.dsmltools.org.
[18] OSG Discovery Tool web site. https://twiki.grid.iu.edu/bin/view/ReleaseDocumentation/OSGStorageDiscoveryTool.
[19] Virtual Data Toolkit web site. http://vdt.cs.wisc.edu.
[20] W3C XQueryX web site. http://www.w3.org/TR/xqueryx.
[21] STAX web site. http://stax.codehaus.org.
[22] Openldap web site. http://www.openldap.org.
[23] Saxon web site. http://saxon.sourceforge.net.